

(12) **EUROPEAN PATENT SPECIFICATION**

(45) Date of publication and mention
of the grant of the patent:
19.04.2000 Bulletin 2000/16

(51) Int. Cl.⁷: G06F 9/44

(21) Application number: 94308623.1

(22) Date of filing: 23.11.1994

(54) **System for speaking hypertext documents such as computerized help files**

System zur Sprachlichen Wiedergabe von Hypertextdokumenten, wie Hilfsdateien

Système de prononciation de documents hypertext, comme des fichiers d'aide

(84) Designated Contracting States:
DE FR GB IT

(30) Priority: 24.11.1993 US 156690

(43) Date of publication of application:
24.05.1995 Bulletin 1995/21

(73) Proprietor:
Canon Information Systems, Inc.
Costa Mesa, CA 92626 (US)

(72) Inventors:
• Tullis, Thomas S.
Laguna Niguel, California 92677 (US)
• Kodimer, Marianne L.
Anahelm, California 92807 (US)

(74) Representative:
Berestford, Keith Denis Lewis et al
BERESFORD & Co.
High Holborn
2-5 Warwick Court
London WC1R 5DJ (GB)

(56) References cited:
EP-A- 0 509 947

- IBM TECHNICAL DISCLOSURE BULLETIN., vol.37, no.4A, April 1994, NEW YORK US pages 669 - 670 I.B.M. CORPORATION 'Method for associating help with host application functions'
- RESEARCH DISCLOSURE, RD263002, 10 March 1986, HAVANT GB ANONYMOUS 'Help system for computer users'
- 24TH ACM/IEEE DESIGN AUTOMATION CONFERENCE, 1987, MIAMI BEACH, FLORIDA, US pages 429 - 435 W. LEE "'?": A context-sensitive help system based on hypertext'
- PROCEEDINGS OF THE HUMAN FACTORS SOCIETY 35TH ANNUAL MEETING, vol.1, 6 September 1991, SAN FRANCISCO, CALIFORNIA, US pages 244 - 248 P.A. SAVAGE ET AL. 'Auditory versus visual presentation of help messages'

Best Available Copy

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

EP 0 654 734 B1

Description

[0001] The present invention concerns a computerized system for speaking hypertext documents, i.e. a document containing embedded links to other locations in the document, such as a computerized help file which contains embedded links to multiple help topics. More particularly, the system concerns a system for providing spoken help information in response to a computer operator's request for help with an application program such as a word processing program or other information processing program.

[0002] With increasing sophistication, complexity and capabilities of computer application programs such as word processing programs and other information processing programs, more and more attention has been focused on providing users of those application programs with on-demand (or on-line) help. It is now common, for example, for computer application programs to designate a special key such as a "F1" key which, when depressed during execution of the application program, will provide the operator with help concerning the application program. Help ordinarily consists of textual information which is stored in a hypertext file of help topics and which is retrieved from that file by a separate computer program, normally called a "help engine", and displayed in text form on the computer screen. Such an operation is disclosed in the Microsoft Professional Toolkit for Visual Basic-Help Compiler Guide 1992 pp 1-71.

[0003] A disadvantage of displaying help information on the computer screen is that the help information obstructs the operator's view of the application program. More particularly, an operator ordinarily requires help while in the middle of executing a software task in conjunction with the application program. Help therefore is requested when the computer's screen is displaying information with which the operator needs help. But when the help information is displayed on the screen, it obstructs the operator's view of the application program with which he needs help, making it difficult to implement the information provided by help. Commonly the operator must read the help text, clear the screen of that text, and then try to remember what it said.

[0004] It is disclosed in an article entitled "Auditory versus Visual Presentation of Help Messages" (Proceedings of the Human Factors Society 35th Annual Meeting, Volume 1, 6th September 1991, San Francisco, CA US, pages 224 to 248, P A Savage *et al*), that help messages can be presented orally on telephone sets as well as presented visually on the telephone sets equipped with display screens.

[0005] In an article entitled "Audio Help for Computer Users" (Research Disclosure RD263002, 10th March 1986, Havant GB, anonymous) there is disclosed a computer based system where a user can access help information visually and/or audibly. The presentation of help information in audio form can supplement or replace the presentation of the same or similar information in the screen.

[0006] In accordance with the first aspect, the present invention provides a method of processing a hypertext help file of topics, at least some of the topics including text and at least one hypertext link to a jump topic, the method comprising the steps of:

- receiving a request for a first topic of the hypertext help file of topics;
- generating speech serially for the text in the first topic;
- accessing a jump topic specified by a said hypertext link in response to a user jump command; and
- generating speech serially for the text in the jump topic;
- characterised by the step of:
- generating speech for the or each hypertext link when encountered during the serial speech generation of the text and generating an aural cue so as to indicate that a hypertext link and not text is being spoken.

[0007] In accordance with a second aspect, the present invention provides an apparatus for processing a hypertext help file of topics, at least some of the topics including text and at least one hypertext link to a jump topic, the apparatus comprising:

- receiving means for receiving a request for a first topic of the hypertext help file of topics;
- speech generation means for serially generating speech for the text in the first topic; and
- accessing means for accessing a jump topic specified by a said hypertext link in response to a received user jump command;
- wherein said speech generation means is adapted to generate speech serially for the text in the jump topic; characterised in that:
- said speech generation means is adapted to generate speech for the or each hypertext link when encountered during the serial speech generation of the text and to generate an aural cue to indicate that the hypertext link and not the text is being spoken.

[0008] Embodiments of the present invention will now be described with reference to the accompanying drawings; in which:

Figure 1 is a perspective view showing the outward appearance of an apparatus according to an embodiment of the invention.

Figure 2 is a block diagram of the Figure 1 apparatus.

Figure 3 is a view for explaining a hierarchically-ordered hypertext file of topics.

5 Figure 4 is a functional block diagram for explaining operation of the visual and spoken help engine according to an embodiment of the invention.

Figure 5 is a representative screen display showing a user interface with the help engine.

Figure 6 is a flow diagram for explaining operation of an embodiment of the invention.

10 [0009] Figure 1 is a view showing the outward appearance of a representative embodiment of the invention. Shown in Figure 1 is computing equipment 10 such as an IBM PC or PC-compatible computer having a Microsoft® Windows windowing system and provided with a display screen 11 such as a color monitor and an internal speaker 12. Computing equipment 10 includes a mass storage device such as computer disk drive 14 for storing data files for application programs, such as bit map image files, text files, sound files, animation files and digital motion video files, in compressed or uncompressed format, as well as for storing the application programs themselves such as a word processing application program or other information processing programs which contain stored program instruction steps by which computing equipment 10 manipulates the data files and presents data in those files to an operator via display screen 11 or speaker 12.

[0010] Although an IBM PC or PC-compatible computer with Windows windowing system is shown, any suitable general or special purpose computer with any appropriate operating system may be used instead.

[0011] Keyboard 15 is connected to computing equipment 10 to permit input of text data and to permit operator selection and manipulation of objects displayed on display screen 11. Likewise, pointing device 16 such as a mouse or the like is connected to permit selection and manipulation of objects on the display screen. Microphone 17 permits inputting audio information.

25 [0012] Computing equipment 10 is also generally provided with various input and output peripherals. Shown here are digital scanner 18 and printer 19, although other devices may also be used such as CD-ROMs, an interface to a network or other external source of data such as a modem or facsimile, a removable floppy disk drive, or a motion video or interface for inputting frames of analog video information.

[0013] In accordance with operator instructions, stored application programs are activated and permit processing and manipulation of data. For example, a word processing program may be activated to permit an operator to create, view, manipulate and print documents which may contain objects such as text objects, bit map objects, computer graphic objects, motion video objects and the like. Other application programs such as spreadsheet programs and similar information processing programs may further be provided on disk 14 for operator selection and use.

[0014] Also stored on disk 14 is a help engine which is an application program used in conjunction with other application programs. More specifically, when help is requested by an operator, the computing equipment instantiates the help engine which provides the operator with on-demand help information. The help engine provides help to the operator in accordance with a hypertext file of help topics which is also stored on disk 14, generally, one hypertext help file for each application program. The help engine is arranged so that it may be set by the operator into different modes to provide visual help and/or spoken help. When providing visual help, the help engine displays text from the hypertext files on display screen 11; likewise, when providing spoken help the help engine speaks the hypertext files over speaker 12.

40 [0015] Figure 2 is a detailed block diagram showing the internal construction of computing equipment 10. As shown in Figure 2, computing equipment 10 includes a central processing unit (CPU) 30 such as an 80386 or a reduced instruction set computer (RISC) interfaced to computer bus 31. Also interfaced to computer bus 31 is display 11, disk 14, keyboard 15 and mouse 16. Scanner and printer interfaces 32 and 34 are provided for interface to scanner 18 and printer 19, respectively, and other interfaces 35 are provided for interface to modems, networks and the like. Audio interface 33 is provided for interfacing with microphone 17 and speaker 12.

[0016] A conventional text-to-speech converter is provided at 37. The text-to-speech converter 37, which may also be implemented in software, interprets text strings sent to it and converts those text strings to audio speech information. The text-to-speech converter 37 provides the audio speech information to speaker 12 for enunciation to the computer operator.

50 [0017] Main memory 36 interfaces to computer bus 31 so as to provide random access memory storage for use by CPU 30. In particular, when executing stored program instruction sequences such as application programs or the help engine, CPU 30 loads those instruction sequences from disk 14 (or other memory storage medium where they are stored) into main memory 36 and executes those stored program instruction sequences out of main memory 36.

55 [0018] Figure 3 is a view for explaining one possible organization of a hypertext file of help topics, and Appendix A is an exemplar of such a file. In this example, the hypertext file is ordered hierarchically with three different hierarchical levels, but it is to be understood that a flat (or single level) hypertext file is also usable. As shown in Figure 3, the hierarchy contains three different levels, namely primary level 40, secondary level 41, and topic level 42. These levels are

for illustrative purposes only and different numbers of levels (including just one) and a different hierarchical structure may be implemented.

[0019] For the three-level structure of Figure 3, plural subjects 44 are indicated at the primary level. Each of those subjects, in turn, has a number of topics 45 included therein as listed at the secondary level 41. Each of those topics, in turn, has text information as indicated at 46.

[0020] Information at topic level 42 is primarily text information designed to give the operator assistance during execution of the application program. Other information may also be included at the topic level and, customarily, information such as bit map images may also be included as desired.

[0021] Information at the primary and secondary levels is primarily designed to define the hierarchical structure of the help file, but it may also include some textual information. Thus, as shown in Appendix A, subject level information which corresponds to secondary level 41, includes text information such as "The following topics are covered in this section" which provides an indication of the hierarchically ordered topics which are to follow.

[0022] Hierarchical links 43a connect primary level 40 to secondary level 41, and hierarchical links 43b connect secondary level 41 to topic level 42. Only some of links 43b have been depicted, but it is understood that a hierarchical link connects each topic heading in secondary level 41 to its associated topic listing in topic level 42.

[0023] Whatever number of hierarchical levels are included, the help file includes imbedded hypertext links which link one area of the file to another. Hypertext links may include links to the beginning of other topics, or they may include links to any location in the file. The hypertext links ordinarily originate from the body of the topics and key into another topic called the "jump" topic. Some hypertext links are depicted at 47a, b and c, but it is to be understood that these depicted links are illustrative only and more or fewer links may be provided as needed.

[0024] Two types of jumps using the links are provided: a return-type jump and a transfer-type jump. A return-type jump allows an operator to use the help engine to temporarily jump to another topic and then return to the jump-off point when the other topic is finished. Typically, a return-type jump is used to obtain a definition of terms, and after the definition is provided the help engine returns to the topic from which the definition was requested. A transfer-type jump, on the other hand, does not return to the topic from which the jump originated, but rather remains in the jumped-to topic. Thus, referring for example to Appendix A, topic number 12 ("Stack") is a return-type jump topic and after processing of that topic is complete, processing reverts to the topic from which the "Stack" topic was jumped from. On the other hand, topic number 4 ("Preventing Stack Overflow") is a transfer-type jump topic and processing does not revert to another topic unless operator intervention is encountered.

[0025] While as described above return-type jumps are used for short, footnote-like explanations, and transfer-type jumps are used for longer, more involved topics, size alone is not determinative, and it is possible to use transfer-type jumps for short topics and return-type jumps for extensive explanations.

[0026] Figure 4 is a functional block diagram of a help engine in accordance with an embodiment of the present invention. As shown in Figure 4, help engine 50 operates generally to retrieve help topics from hypertext help file 40 and, when in the spoken help mode, to speak them via text-to-speech interface 37 and, when in a visual mode, to display them via display 11. When in both a spoken and visual mode, both of these actions occur. The topics that are retrieved from the hypertext file are selected by the operator in accordance with manipulation of keyboard 15 or graphic user interface 51. The initial topic, however, may be set by the application program which instantiates the help engine. In more detail, when help is requested from within the application program 52, the application program instantiates help engine 50 and at the same time designates the current context of the application. The context, which may simply be a context number, is related to the current focus of the application, for example, a particular task that the operator is executing. The context is passed to context detector 53 within the help engine which retrieves the appropriate topic via topic retriever 54. The retrieved topic is spoken and/or displayed in accordance with the current mode of the help engine.

[0027] As topics are retrieved by topic retriever 54 and spoken and/or displayed, both hierarchical and hypertext jump topics are detected and stored in lists 55 and 56, respectively. In the spoken help mode, the hierarchical and hypertext jump topics are spoken with an aural cue to the operator such as by speaking with a voice that differs from the remainder of the topic; in the visual mode the hypertext and hierarchical jump topics are displayed with visual cues, for example, by single or double underlining or with different colors or fonts.

[0028] Figure 5 is a view of a computer-generated control panel displayable in a graphical user interface in connection with the help engine. Using controls indicated at 51a, it is possible to jump to one of the hypertext or hierarchical jump topics in lists 55 and 56, as described in more detail below. Using controls indicated at 51b, the operator may control the processing of text within a topic, for example, by skipping backward or forward through sentences or paragraphs. Using controls indicated at 51c, it is possible to change to different hierarchical levels in response to which the help engine speaks and/or displays the index at the new level. Finally, using controls at 51d, it is possible to change the text-to-speech parameters such as the speed or volume of the text-to-speech conversion.

[0029] In some applications, the control panel of Figure 5 need not be displayed, and it is possible to provide user control over the help engine via other means such as keyboard control or speech recognition of spoken commands.

[0030] Figure 6 is a flow diagram illustrating operation of the help engine according to an embodiment of the inven-

tion.

[0031] In step S601, an application program such as an information processing program is executed, and execution of that program continues until step S602 in which a request for help is issued by the operator. In step S603, the help engine is instantiated. The help engine may be instantiated in either visual or spoken mode, or both modes, as set by the operator in step S604. In step S605, the help engine selects the initial help topic. The initial selection may be made in accordance with context passed to the help engine by the application program, or it may be selected by other means, for example, by operator selection or by a default setting such as the primary level 40 in the hierarchical help file.

[0032] In step S606, in accordance with the selected spoken/visual mode, the help engine begins speaking and/or displaying the selected help topic. If a hierarchical link is encountered while speaking the help topic (step S607), then in step S608 the hierarchical link is stored in list 55 and in step S609 an aural cue is given to the operator by changing the voice parameters for the text-to-speech converter so as to speak the hierarchical link in a voice (pitch, intonation, etc.) distinguishable from the remainder of the topic. Likewise, if a hypertext link is encountered (step S610), then in step S611 the hypertext link is stored in list 56 and in step S612 an aural cue is given to the operator by changing voice parameters of the text-to-speech converter so as to speak the hypertext link in a voice that is different than the remainder of the topic.

[0033] In step S613, the help engine determines whether a user command has been entered via keyboard 15 or graphical user interface 51. If no user commands have been entered, then flow advances to step S614 where the help engine determines if an end of topic has been encountered. If the end of topic has not been encountered then flow returns to step S606 where the help engine continues to speak and/or display the topic. On the other hand, if end of topic has been reached, then flow returns to step S613 where the help engine waits for operator input. Alternatively, though not shown, if end of topic has been reached, the help engine may begin to process the next sequential topic and continue on through to the end of all the topics in the current hierarchical level, whereafter flow returns to step S613.

[0034] When an operator command is detected in step S613 then flow advances to steps S615 through S631 where the help engine analyzes the operator command and takes the appropriate action. More specifically, if in step S615 the help engine detects a command to jump to a hypertext topic then flow advances to step S616 in which the help engine jumps to the last hypertext topic stored in list 56. If the hypertext jump is a transfer-type jump, then in step S617 the help engine changes the current topic and flow returns to step S606 to speak the topic. On the other hand, if the jump is a return-type jump, then in step S618 the help engine speaks the selected topic and then reverts back to the jumped-from topic to complete speaking that topic. Preferably, for a return-type jump, the voice in which the topic is spoken is the same as the voice in which the topic name was enunciated to the operator. It is also preferable to return to the jumped-from topic not from the point at which the jump was made but rather from the beginning of the sentence from which the jump was made.

[0035] If the help engine did not encounter a hypertext jump command in step S615, then flow advances to step S620 in which the help engine determines whether a hierarchical jump command has been issued. If a hierarchical jump command has been issued, then flow advances to steps S621 through steps S626 so as to process the hierarchical jump command. More specifically, if in step S621 the operator selects a particular topic that is being read from a list of hierarchical topics, then in step S622 the topic is changed to the selected topic and flow returns to step S606 to speak/display the topic. Selection in step S621 may be made in accordance with an operator indication to select the most recently read topic from the list (for example, by clicking a button on mouse 16) or selection may be made in accordance with a number associated with a list of topics. In this latter case, topics are spoken/displayed sequentially from a list of hierarchical topics and the operator may select any one of those topics at any time (and not just immediately after the topic is spoken and/or displayed) by entering the appropriate number of the topic.

[0036] If the operator is not selecting a hierarchical topic in step S621, but in step S623 is selecting the next and/or previous topic via the graphical user interface, then in step S624 the next or previous topic, as stored in list 55, is selected in step S624 and flow returns to step S606 where the topic is spoken and/or displayed.

[0037] If next/previous topic processing is not selected in step S623 but previous level processing is selected in step S625, then in accordance with the hierarchical list stored in list 55 the previous topic level is selected in step S626 and flow returns to step S606 to speak the list of topics in that level.

[0038] If a hierarchical jump command is not received in step S620, then in step S627 if the operator has input topic repositioning commands then those commands (step S628) are executed and flow returns to step S606 to continue speaking the topic from the repositioned position.

[0039] If a topic repositioning command is not received in step S627, but in step S629 a command to close help is received, then in step S630 the help engine is closed down and in step S631 CPU control reverts to the application program 62 which originally instantiated the help engine.

[0040] The following is an example of help engine processing in accordance with the hierarchically ordered hypertext topic file shown in Appendix A. The purpose of this example is to show the effects of flow processing in accordance with the navigation commands provided by the graphical user interface 51 and to show how differing voice output can provide an operator with aural cues which define the document structure and which change in accordance with that

structure. For purposes of this example, different voices are indicated by the following different text styles:

Key

5	<hr/>	
	<user command>	
	body of topic	Voice 1
	<i>HIERARCHICAL JUMP TOPIC</i>	VOICE 2
10	<u>transfer-type hierarchical jump topic</u>	Voice 3
	<u>return-type hypertext jump topic</u> → <u>Voice 4</u>	
15	<hr/>	

[0041] For purposes of this example, it is assumed that the help engine is instantiated to begin speaking the Appendix A help file commencing from the primary level hierarchical structure. As described above, however, the help engine may be instantiated to begin speaking the help file from a topic defined by the context of the application program which instantiates the help engine.

[0042] Using the above key, the help engine commences speaking the help file commencing with topic number 1, primary hierarchical level, as follows (step numbers from Figure 6 have been inserted at the left margin):

25

30

35

40

45

50

55

5

S607 to S609	<ol style="list-style-type: none"> 1. GENERAL INFORMATION ABOUT MICROSOFT WINDOWS VERSION 3.1 2. USING MULTIMEDIA EXTENSIONS WITH WINDOWS VERSION 3.1 3. USING SPECIFIC DISPLAYS WITH WINDOWS VERSION 3.1
--------------------	--

10

S621	<"Select from menu: "1">
------	--------------------------

15

20

25

S606 etc.	<p>General information about Microsoft Windows version 3.1. This document contains important information that is not included in the Microsoft Windows User's Guide or in online Help. If you plan on using a terminate-and-stay resident (TSR) program with Windows, please read the SETUP.TXT online document first. It contains important information about running TSR programs with Windows version 3.1 and Windows Setup. The following topics are covered in this section.</p>
--------------	---

30

S607 to S609	<ol style="list-style-type: none"> 1. Running Windows with an Operating System Other Than MS-DOS 2. Preventing Stack Overflow
--------------------	---

35

S621	<"Jump" command>
------	------------------

40

S610 to S612	<p>Preventing Stack Overflow. If, while setting up Windows, Setup detects hardware or software that require a certain <u>stack</u> size</p>
--------------------	---

45

S615	<"Jump" command>
------	------------------

50

S616 to S619	<p><u>Stacks are temporary data structures that MS-DOS and applications use for processing hardware events.</u></p>
--------------------	---

55

S606 etc. If, while setting up Windows, Setup detects hardware or software that require a certain stack size, Setup will place the following command line in your CONFIG.SYS file: stacks=9,256. This should be sufficient most of the time. However, if you receive the Internal Stack Overflow message when running Windows in 386 enhanced mode, or if your system fails for no apparent reason when running Windows in standard mode (especially if you are setting up or moving the mouse), first try increasing the second number on the stacks=command line (for example 256). If that doesn't work, try increasing the first number (for example, 9). For more information about the stacks setting and modifying the CONFIG.SYS file, see your MS-DOS documentation.

S623 <"previous topic" command>

S607 2. Preventing Stack Overflow
to 3. Using 32-Bit Disk Access
S609 4. Using Memory Managers

S621 <Select from menu "4">

S606 etc. This section describes problems you may encounter using memory managers with Windows version 3.1. The following topics will be covered:

S610 a. Solving Memory Conflicts by Using
to MONOUMB.386 or MONOUMB2.386
S612

S617 <"Jump" command>

S606 etc. Solving Memory Conflicts by Using MONOUMB.386 or MONOUMB2.386. If you encounter the following message when starting Windows in 386 enhanced mode, your display driver may be accessing the monochrome address range (B000-B7FF), which prevents the memory manager from using this range for upper memory blocks.

S610 To solve this problem, try installing
to Monumb2.386 on your system.
S612

	S617 <"Jump" command>
5 10 15 20 25 30 35	<div data-bbox="349 325 1234 1218"> <p>S606 etc. To install Monoumb2.386 do the following:</p> <ol style="list-style-type: none"> Copy and expand the MONOUMB2.38_ file that is on your Windows disk to your Windows SYSTEM directory by typing the following at the MS-DOS prompt: <code>expand monoumb2.38_ c:\windows\systems</code> Add the following setting to the [386Enh] section in the SYSTEM.INI file: <code>device=monoumb2.386</code> Start Windows. <p>Note: MONOUMB2.386 may not work with some memory managers, such as EMM386.EXE. In this case, you can try using MONOUMB.386, provided with the Windows Driver Library. To obtain a copy of the Windows Driver Library. To obtain a copy of the Windows Driver Library, contact Microsoft.</p> <p>You can also exclude the address region B000-B7FF, which specifies that the memory manager should not try to use this address range for upper memory blocks. For information about excluding specific address ranges, see <i>solving memory conflicts by excluding an address range</i>.</p> <p>S610 to S612</p> </div>
40	<div data-bbox="349 1270 1234 1354"> <p>S613 to S614 <stop until further operator input></p> </div>

45 Instead of this last step ("<stop until further operator input>"), and as mentioned above, it is possible in the alternative to continue speaking with the next sequential topic and in the absence of operator input continue speaking until all topics in the current hierarchical level have been spoken.

[0043] The above example is representative only, and the scope of the invention should not be measured against the example but rather should be measured against the appended claims.

50

55

APPENDIX A

The following is one possible example of how a hypertext document, here, a hierarchically-ordered hypertext document, might be structured to be used by the spoken/visual help engine of the invention. There are three hierarchical levels: primary level, secondary level, and topic level (topic #'s 1 to 10). Within the body of the document, transfer-type Jump Topics are denoted by a single underline and return-type Jump Topics are denoted by a double underline.

Note. Though the Contents contains three subject headings corresponding to three subjects at the secondary level, only the first one (*General Information About Microsoft Windows Version 3.1*) is included in this appendix.

CONTENTS

1. General Information about Microsoft Windows
Version 3.1
2. Using Multimedia Extensions with Windows
Version 3.1
3. Using Specific Displays with Windows Version
3.1

Primary Level End

- # 1. General Information About Microsoft Windows
- ## Version 3.1

This document contains important information that is not included in the Microsoft Windows User's Guide or in on-line Help.

If you plan on using a terminate-and-stay resident (TSR) program with Windows, please read the SETUP.TXT online document first. It contains important information about running TSR programs with Windows version 3.1 and Windows Setup.

The following topics are covered in this section:

1. Running Windows With An Operating System Other Than MS DOS
2. Preventing Stack Overflow
3. Using 32-Bit Disk Access
4. Using Memory Managers

Subject #1 End

-
1. Running Windows with an Operating System Other Than MS DOS

Microsoft Windows and MS-DOS work together as an integrated system. They were designed together and extensively tested together on a wide variety of computers and hardware configurations. Running Windows version 3.1 on an operating system other than MS-DOS could cause unexpected results or poor performance.

Topic #1 End

-
2. Preventing Stack Overflow

If, while setting up Windows, Setup detects hardware or software that require a certain size for stacks, Setup will place the following command line in your CONFIG.SYS file: stacks=9,256

This should be sufficient most of the time. However, if you receive the Internal Stack Overflow message when running Windows in 386 enhanced mode, or if your system fails for no apparent reason when running Windows in standard mode (especially if you are setting up or moving the

mouse), first try increasing the second number on the stacks=command line (for example, 256)

If that doesn't work, try increasing the first number (for example, 9). For more information about the stacks setting and modifying the CONFIG.SYS file, see your MS-DOS documentation.

Topic #2 End

3. Using 32-Bit Disk Access

By default, 32-bit disk access is turned off to prevent disk errors on some battery-powered portable computers. However, 32-bit disk access may run correctly with some battery-powered portable computers, such as the Zenith MasterSport SL. If you want to try using 32-bit disk access, select the 32-bit disk access check box in the Virtual Memory dialog box (choose the 386 Enhanced option in Control Panel).

Topic #3 End

4. Using Memory Mangers

This section describes problems you may encounter using memory managers with Windows version 3.1. The following topics are covered.

- a. Solving Memory Conflicts by Using MONOUMB.386 or MONOUMB2.386
- b. Solving Memory Conflicts by Excluding An Address Range
- c. Using 386 MAX

Topic #4 End

a. *Solving Memory Conflicts by Using MONOUMB.386 or MONOUMB2.386*

If you encounter the following message when starting Windows in 386 enhanced mode your display driver may be accessing the monochrome address range (B000-B7FF), which prevents the memory

manager from using this range for upper memory blocks:

To solve this problem, try installing MONOUMB2.386 on your system. MONOUMB2.386 is a device driver provided with Windows that allows certain memory managers to use the monochrome address range for upper memory blocks, even if your display driver is accessing this range.

Topic #5 End

b. Solving Memory Conflicts by Excluding an Address Range

If you encounter the following message when starting Windows in the 386 enhanced mode, and the address specified is not B000, you must exclude the address range.

If the address specified is B000, you can try using MONOUMB2.386 or MONOUMB.386. If you are using QEMM, you need to include the X= option on the device=qemm386.sys command line in your CONFIG.SYS file. For example, to exclude the address range C000-C7FF, you would specify the following:

device=qemm386.sys X=C000-C7FF

Topic #6 End

c. Using 386MAX

If you are running 386MAX with Windows version 3.1, note the following:

- Do not use options that limit the EMS swap region in the upper memory area. If you include exclude=options on the 386max or bluemax command line or in the .PRO file (usually called 386MAX.PRO), make sure that the address range specified does not extend beyond A000. See the example on address range boundaries.

- Do not load SMARTDrive when QCache is running.

If you update an earlier version of 386MAX to version 6.0, the 386MAX version 6.0 setup program may not remove the LOAD=WINDOWS.LOD line from your .PRO file. You can remove this line manually. It is not needed and removing it will free up a small amount of memory.

Topic #7 End

installing MONOUMB2.386

To install MONOUMB2.386 do the following.

1. Copy and expand the MONOUMB2.38_ file that is on your Windows disk to your Windows SYSTEM directory by typing the following at the MS-DOS prompt:

expand monoumb2.38_ c:\windows\system

2. Add the following setting to the [386Enh] section in the SYSTEM.INI file:

device=monoumb2.386

3. Start Windows.

Note: MONOUMB2.386 may not work with some memory mangers, such as EMM386.EXE. In this case, you can try using MONOUMB.386, provided with the Windows Driver Library. To obtain a copy of the Windows Driver Library, contact Microsoft.

You can also exclude the address region B000-B7FF, which specifies that the memory manager should not try to use this address ranges, see the following topic.

Topic #8 End

Example on address range boundaries

For example, exclude=1800-A000 is acceptable, but exclude+1800-B800 is not. If the address range specified by using the exclude=option

extends beyond A000, Windows may not run properly in
 386 enhanced mode. If you must exclude an address
 range above A000, use the RAM=option instead. For
 more information about this and other options, see
 your 386MAX documentation.

Topic #9 End

Stack

Stacks are temporary data structures that
 MS-DOS and applications use for processing hardware
 events.

Topic #10 End

Claims

1. A method of processing a hypertext help file of topics, at least some of the topics including text and at least one hypertext link to a jump topic, the method comprising the steps of:

receiving a request for a first topic of the hypertext help file of topics;
 generating speech serially for the text in the first topic;
 accessing a jump topic specified by a said hypertext link in response to a user jump command; and
 generating speech serially for the text in the jump topic;
 characterised by the step of:
 generating speech for the or each hypertext link when encountered during the serial speech generation for the
 text and generating an aural cue so as to indicate that a hypertext link and not text is being spoken.

2. A method according to claim 1, wherein the aural cue comprises the generation of speech for the or each hypertext link in a second voice which is different from a first voice generated for the text in the first or jump topic.

3. A method according to claim 1 or claim 2, including optionally displaying the text and the or each hypertext link of the topic for which speech is being generated.

4. A method according to claim 3, wherein the steps of claim 3 are implemented by a help engine, the method further comprising the steps of:

executing an application program which includes operator-invokable commands to execute the help engine
 which is settable to provide any one of speech generated help, displayed help, and both speech generated and
 displayed help; and
 instantiating the help engine in response to operator commands.

5. A method according to claim 4, further comprising the step of passing context of the application program to the help engine, where speech is generated for topics and optionally topics are displayed commencing with the context passed from the application program.

6. A method according to claim 4 or claim 5, further comprising the step of displaying a graphic user interface by which the help engine is controlled.

7. A method according to claim 6, wherein the hypertext help file is arranged hierarchically, and wherein the graphic user interface includes a set of processing commands to process the hierarchical levels of the hypertext file.
- 5 8. A method according to claim 7, wherein the set of processing commands includes commands to jump to different topics.
9. A method according to claim 7, wherein the set of processing commands includes commands to jump to different levels.
- 10 10. A method according to claim 7, wherein the set of processing commands includes commands to reposition the speech generator and optional display of the topic.
11. A method according to any preceding claim, further comprising the step of storing one of said at least one hypertext link.
- 15 12. A method according to claim 11, further comprising the step of accessing the topic specified by the stored link in response to user jump commands and generating speech for the jump topic.
13. A method according to claim 12, further comprising the step of returning to the jumped-from topic at the conclusion of the jump topic.
- 20 14. A method according to any preceding claim, wherein the hypertext file is arranged hierarchically, and further comprising the step of detecting and storing the hierarchical structure of topics.
- 25 15. A method according to claim 14, further comprising the step of jumping to different topics in the stored hierarchical structure in accordance with operator commands.
16. A method according to claim 14, further comprising the step of generating speech for the stored hierarchical structure of topics.
- 30 17. A method according to claim 16, further comprising the step of jumping to a selected one of the spoken topics.
18. A method according to any preceding claim, further comprising the steps of, while in the jump topic accessing a second jump topic specified by another of said at least one hypertext link in response to a second user jump command.
- 35 19. A method according to any preceding claim, further comprising the step of returning to the first topic at the conclusion of the generation of speech for the jump topic.
- 40 20. A method according to any preceding claim, further comprising the step of detecting context wherein the first topic is context dependent.
21. A method according to any preceding claim, further comprising the step of displaying a graphical user interface by which an operator may issue jump commands.
- 45 22. A method according to claim 21, wherein said graphical user interface receives commands to reposition speech generation for the topic.
- 50 23. Apparatus for processing a hypertext help file of topics, at least some of the topics including text and at least one hypertext link to a jump topic, the apparatus comprising:

receiving means for receiving a request for a first topic of the hypertext help file of topics;
speech generation means for serially generating speech for the text in the first topic; and
accessing means for accessing a jump topic specified by a said hypertext link in response to a received user
55 jump command;
wherein said speech generation means is adapted to generate speech serially for the text in the jump topic;
characterised in that:
said speech generation means is adapted to generate speech for the or each hypertext link when encountered

during the serial speech generation of the text and to generate an aural cue to indicate that a hypertext link and not text is being spoken.

24. Apparatus according to claim 23, wherein said speech generation means is adapted to generate speech for the or each hypertext link in a second voice which is different from a first voice generated for the text in the first or jump topic, so as to provide the aural cue to indicate that a hypertext link and not text is being spoken.
25. Apparatus according to claim 24, comprising a processor for executing processor implementable instructions, said processor being adapted to implement processor implementable instructions to execute an application program which includes operator invocable commands to execute a help engine which is settable to provide any one of speech generated help, displayed help, and both speech generated and displayed help, said help engine comprising said receiving means, said speech generation means, said accessing means, and said display control means.
26. Apparatus according to claim 25, wherein said processor is adapted to pass context of the application program to the help engine, and the help engine is adapted to commence speech generation for topics and optionally display topics with context passed from the application program.
27. Apparatus according to claim 25 or claim 26, wherein said processor is adapted to generate a graphic user interface for the control of the help engine.
28. Apparatus according to claim 27, wherein the hypertext help file is arranged hierarchically, and the processor is adapted to implement the graphic user interface to include a set of processing commands to process the hierarchical levels of the hypertext files.
29. Apparatus according to claim 28, wherein the processor is adapted to implement the graphical user interface to include commands to jump to different topics.
30. Apparatus according to claim 28, wherein the processor is adapted to implement the graphical user interface to include commands to jump to different levels.
31. Apparatus according to claim 28, wherein the processor is adapted to implement the graphical user interface to include commands to reposition the speech generation and optional display of the topic.
32. Apparatus according to any one of claims 23 to 31, further comprising means for storing one of said at least one hypertext link.
33. Apparatus according to claim 32, wherein said help engine is adapted to access the topics specified by the stored link in response to user jump commands and to generate speech for the jump topic.
34. Apparatus according to claim 33, wherein said help engine is adapted to return to the jumped-from topic at the conclusion of the jump topic.
35. Apparatus according to any one of claims 23 to 34, wherein the hypertext file is arranged hierarchically, and said help engine is adapted to detect and store the hierarchical structure of topics.
36. Apparatus according to claim 35, wherein said help engine is adapted to jump to different topics in the stored hierarchical structure in accordance with operator commands.
37. Apparatus according to claim 36, wherein said help engine is adapted to generate speech for the stored hierarchical structure of topics.
38. Apparatus according to claim 37, wherein said speech engine is adapted to jump to a selected one of the spoken topics.
39. Apparatus according to any one of claims 23 to 38, wherein said speech engine is adapted to, while in the jump topic, access a second jump topic specified by another of said at least one hypertext link in response to a second user jump command.

40. Apparatus according to any one of claims 23 to 39, wherein said help engine is adapted to return to the first topic at the conclusion of the generation of speech for the jump topic.

41. Apparatus according to any one of claims 23 to 40, wherein said help engine is adapted to detect context wherein the first topic is context dependent.

42. Apparatus according to any one of claims 23 to 41, wherein said processor is adapted to display a graphical user interface by which an operator may issue jump commands.

43. Apparatus according to claim 43, wherein said processor is adapted to implement said graphical user interface to receive commands to reposition speech generation for the topic.

44. A storage medium storing program instructions for controlling a processor to carry out the method of any one of claims 1 to 22.

45. Processor implementable program instructions for controlling a processor to carry out the method of any one of claims 1 to 22.

Patentansprüche

1. Verfahren zur Verarbeitung einer Hypertext-Hilfedatei, die thematische Einzelpunkte enthält, von denen wenigstens einige der Einzelpunkte Text und wenigstens eine Hypertext-Verknüpfung zu einem Sprungpunkt (anzuspringenden Einzelpunkt) einschließen, wobei das Verfahren folgende Schritte umfaßt:

Empfangen einer Anforderung für einen ersten Einzelpunkt der Hypertext-Hilfedatei der Einzelpunkte,

Erzeugen von Sprache sequentiell für den Text im ersten Einzelpunkt,

Zugriff auf einen Sprungpunkt, der durch die Hypertext-Verknüpfung bezeichnet wird, in Reaktion auf ein vom Anwender erteiltes Sprungkommando und

Erzeugen von Sprache sequentiell für den Text im Sprungpunkt, gekennzeichnet durch einen Schritt, bei dem

Sprache für die oder jede Hypertext-Verknüpfung, sobald sie während der sequentiellen Spracherzeugung für den Text auftritt, und ein klanglich unterscheidbarer Hinweis, um anzuzeigen, daß gerade eine Hypertext-Verknüpfung und nicht Text gesprochen wird, erzeugt wird.

2. Verfahren nach Anspruch 1, worin der klanglich unterscheidbare Hinweis die Erzeugung von Sprache für die oder jede Hypertext-Verknüpfung in einer zweiten Stimme, die sich von der ersten Stimme, die für den Text im ersten Einzelpunkt oder im Sprungpunkt erzeugt wird, unterscheidet, umfaßt.

3. Verfahren nach Anspruch 1 und Anspruch 2, das gegebenenfalls die Anzeige des Textes und der oder jeder Hypertext-Verknüpfung des Punktes, für den Sprache erzeugt wird, einschließt.

4. Verfahren nach Anspruch 3, worin die Schritte aus Anspruch 3 durch eine Hilfemaschine implementiert werden, wobei das Verfahren weiter folgende Schritte umfaßt:

Ausführen eines Anwendungsprogrammes, das durch den Bediener aufrufbare Kommandos einschließt, um die Hilfemaschine auszuführen, die einstellbar ist auf die Bereitstellung von entweder nur spracherzeugender Hilfe, angezeigter Hilfe oder sowohl spracherzeugender als auch angezeigter Hilfe und

Aufrufen der Hilfemaschine in Reaktion auf Bedienerkommandos.

5. Verfahren nach Anspruch 4, das weiter den Schritt umfaßt, daß der Kontext des Anwendungsprogrammes an die Hilfemaschine durchgereicht wird, mittels derer Sprache für die Einzelpunkte erzeugt wird und gegebenenfalls die Einzelpunkte angezeigt werden, beginnend mit dem Kontext, der vom Anwendungsprogramm übergeben wird.

6. Verfahren nach Anspruch 4 oder Anspruch 5, das weiter den Schritt umfaßt, daß eine grafische Benutzerschnittstelle angezeigt wird, durch welche die Hilfemaschine gesteuert wird.
- 5 7. Verfahren nach Anspruch 6, worin die Hypertext-Hilfdatei hierarchisch angeordnet ist und worin die grafische Benutzerschnittstelle einen Satz von Verarbeitungskommandos zum Verarbeiten der hierarchischen Stufen der Hypertextdatei einschließt.
8. Verfahren nach Anspruch 7, worin der Satz der Verarbeitungskommandos Kommandos zum Anspringen von verschiedenen Einzelpunkten umfaßt.
- 10 9. Verfahren nach Anspruch 7, worin der Satz der Verarbeitungskommandos Kommandos zum Anspringen von verschiedenen Ebenen einschließt.
10. Verfahren nach Anspruch 7, worin der Satz von Verarbeitungskommandos Kommandos zum Neupositionieren des Sprachgenerators und der optionalen An. zeige des Einzelpunktes einschließt.
- 15 11. Verfahren nach einem der vorhergehenden Ansprüche, das weiter den Schritt umfaßt, daß eine der Wenigstens einen Hypertext-Verknüpfungen gespeichert wird.
- 20 12. Verfahren nach Anspruch 11, das weiter den Schritt umfaßt, daß der Einzelpunkt, der durch die gespeicherte Verknüpfung angegeben wird, in Reaktion auf ein Anwender-Sprungkommando angesprungen wird und Sprache für den Sprungpunkt erzeugt wird.
- 25 13. Verfahren nach Anspruch 12, das weiter den Schritt umfaßt, daß zum Punkt, von dem weggesprungen wurde, zurückgekehrt wird, wenn der Sprungpunkt geschlossen wird.
14. Verfahren nach einem der vorhergehenden Ansprüche, worin die Hypertext-datei hierarchisch angeordnet ist, und das weiter den Schritt umfaßt, daß die hierarchische Struktur der Einzelpunkte festgestellt und gespeichert wird.
- 30 15. Verfahren nach Anspruch 14, das weiter den Schritt umfaßt, daß verschiedene Einzelpunkte in der gespeicherten hierarchischen Struktur gemäß Bedienerkommandos angesprungen werden.
16. Verfahren nach Anspruch 14, das weiter den Schritt umfaßt, daß Sprache für die gespeicherte hierarchische Struktur der Einzelpunkte erzeugt wird.
- 35 17. Verfahren nach Anspruch 16, das weiter den Schritt umfaßt, daß ein ausgewählter Punkt der vorgelesenen Einzelpunkte angesprungen wird.
- 40 18. Verfahren nach einem der vorhergehenden Ansprüche, das weiter die Schritte umfaßt, daß, während man sich im ersten Sprungpunkt befindet, ein zweiter Sprungpunkt, der durch eine andere der wenigstens einen Hypertext-Verknüpfungen bezeichnet wird, in Reaktion auf ein zweites Bediener-Sprungkommando zugegriffen wird.
19. Verfahren nach einem der vorhergehenden Ansprüche, das weiter den Schritt umfaßt, daß zum ersten Einzelpunkt zurückgekehrt wird, nachdem die Erzeugung der Sprache für den Sprungpunkt abgeschlossen ist.
- 45 20. Verfahren nach einem der vorhergehenden Ansprüche, das weiter den Schritt umfaßt, daß der Kontext festgestellt wird, in dem der erste Einzelpunkt kontextabhängig ist.
21. Verfahren nach einem der vorhergehenden Ansprüche, das weiter den Schritt umfaßt, daß eine grafische Benutzerschnittstelle angezeigt wird, durch die ein Bediener Sprungkommandos absetzen kann.
- 50 22. Verfahren nach Anspruch 21, worin die grafische Benutzerschnittstelle Kommandos zur Neupositionierung der Spracherzeugung für den Einzelpunkt empfängt.
- 55 23. Vorrichtung zur Verarbeitung einer Hypertext-Hilfdatei, die thematische Einzelpunkte enthält, von denen wenigstens einige der Einzelpunkte Text und wenigstens eine Hypertext-Verknüpfung zu einem Sprungpunkt (anzuspringenden Einzelpunkt) einschließen, wobei die Vorrichtung folgendes umfaßt:

Eine Empfangseinrichtung zum Empfang einer Anforderung für einen ersten Einzelpunkt der Hypertext-Hilfdatei der Einzelpunkte,

eine Spracherzeugungseinrichtung zum sequentiellen Erzeugen von Sprache für den Text im ersten Einzelpunkt und

eine Zugriffseinrichtung zum Zugreifen auf einen Sprungpunkt, der durch die Hypertext-Verknüpfung bezeichnet wird, in Reaktion auf ein empfangenes Benutzer-Sprungkommando, worin die Spracherzeugungseinrichtung so eingerichtet ist, daß sie sequentiell Sprache erzeugt für den Text im Sprungpunkt, dadurch gekennzeichnet daß:

Die Spracherzeugungseinrichtung so eingerichtet ist, daß sie Sprache für die oder jede Hypertext-Verknüpfung, sobald sie während der sequentiellen Spracherzeugung für den Text auftritt, und einen klanglich unterscheidbaren Hinweis, um anzuzeigen, daß gerade eine Hypertext-Verknüpfung und nicht Text gesprochen wird, erzeugt

24. Vorrichtung nach Anspruch 23, worin die Spracherzeugungseinrichtung so eingerichtet ist, daß sie Sprache für die oder jede Hypertext-Verknüpfung in einer zweiten Stimme, die sich von der ersten Stimme, die für den Text im ersten Einzelpunkt oder im Sprungpunkt erzeugt wird, unterscheidet, erzeugt, so daß sie einen klanglich unterscheidbaren Hinweis, um anzuzeigen, daß gerade eine Hypertext-Verknüpfung und nicht Text gesprochen wird, bereitstellt.

25. Vorrichtung nach Anspruch 24, umfassend einen Prozessor zum Ausführen von prozessorimplementierbaren Anweisungen, wobei der Prozessor so eingerichtet ist, daß er prozessorimplementierbare Anweisungen implementiert zum Ausführen eines Anwendungsprogrammes, das durch den Bediener aufrufbare Kommandos einschließt: um eine Hilfemaschine auszuführen, die einstellbar ist auf die Bereitstellung von spracherzeugender Hilfe, angezeigter Hilfe oder sowohl spracherzeugender als auch angezeigter Hilfe, wobei die Hilfemaschine die Empfangseinrichtung, die Spracherzeugungseinrichtung, die Zugriffseinrichtung und die Anzeigesteuereinrichtung umfaßt.

26. Vorrichtung nach Anspruch 25, worin der Prozessor so eingerichtet ist, daß er den Kontext des Anwendungsprogrammes zur Hilfemaschine weiterreicht, und die Hilfemaschine so eingerichtet ist, daß sie die Spracherzeugung für Einzelpunkte und gegebenenfalls das Anzeigen von Einzelpunkten mit dem Kontext, der vom Anwendungsprogramm übergeben wurde, beginnt.

27. Vorrichtung nach Anspruch 25 oder Anspruch 26, worin der Prozessor so eingerichtet ist, daß er eine grafische Benutzerschnittstelle zum Erzeugen zur Steuerung der Hilfemaschine erzeugt.

28. Vorrichtung nach Anspruch 27, worin die Hypertext-Hilfdatei hierarchisch angeordnet ist und der Prozessor so eingerichtet ist, daß er die grafische Benutzerschnittstelle so implementiert, daß sie einen Satz von Bearbeitungskommandos einschließt, um die hierarchischen Ebenen der Hypertextdatei zu bearbeiten.

29. Vorrichtung nach Anspruch 28, worin der Prozessor so eingerichtet ist, daß er die grafische Benutzerschnittstelle so implementiert, daß sie Kommandos zum Springen zu verschiedenen Punkten einschließt.

30. Vorrichtung nach Anspruch 28, worin der Prozessor so eingerichtet ist, daß er die grafische Benutzerschnittstelle so implementiert, daß sie Kommandos zum Springen zu verschiedenen Ebenen einschließt.

31. Vorrichtung nach Anspruch 28, worin der Prozessor so eingerichtet ist, daß er die grafische Benutzerschnittstelle so implementiert, daß sie Kommandos zum Neupositionieren der Spracherzeugung und der optionalen Anzeige des Einzelpunktes einschließt.

32. Vorrichtung nach einem der Ansprüche 23 bis 31, die weiter eine Einrichtung zum Speichern der wenigstens einen Hypertext-Verknüpfung umfaßt.

33. Vorrichtung nach Anspruch 32, worin die Hilfemaschine so eingerichtet ist, daß sie die Punkte, die durch die gespeicherte Verknüpfung bezeichnet werden, in Reaktion auf Anwender-Sprungkommandos zugreift und Spra-

che für den angesprungenen Punkt erzeugt.

34. Vorrichtung nach Anspruch 33 worin die Hilfemaschine so eingerichtet ist, daß sie zum Einzelpunkt, von dem weg-
gesprungen wurde, nach Abschluß des Sprungpunktes zurückkehrt.
35. Vorrichtung nach einem der Ansprüche 23 bis 34, worin die Hypertextdatei hierarchisch angeordnet ist und die Hil-
femaschine so eingerichtet ist, daß sie die hierarchische Struktur der Einzelpunkte feststellt und speichert.
36. Vorrichtung nach Anspruch 35, worin die Hilfemaschine so eingerichtet ist, daß sie nach Maßgabe von Komman-
dos des Operators zu verschiedenen Punkten in der gespeicherten hierarchischen Struktur springt.
37. Vorrichtung nach Anspruch 36, worin die Hilfemaschine so eingerichtet ist, daß sie Sprache erzeugt für die abge-
speicherte hierarchische Struktur der Einzelpunkte.
38. Vorrichtung nach Anspruch 37, worin die Sprachmaschine so eingerichtet ist, daß sie zu einem ausgewählten der
gesprochenen Punkte springt.
39. Vorrichtung nach einem der Ansprüche 23 bis 38, worin die Sprachmaschine so eingerichtet ist, daß sie, während
sie sich in einem Sprungpunkt befindet, einen zweiten Sprungpunkt, der durch eine andere der wenigstens einen
Hypertext-Verknüpfungen bezeichnet wird, in Reaktion auf ein zweites Anwender-Sprungkommando zugreift.
40. Vorrichtung nach einem der Ansprüche 23 bis 39, worin die Hilfemaschine so eingerichtet ist, daß sie zum ersten
Punkt zurückkehrt, wenn die Spracherzeugung für den Sprungpunkt abgeschlossen ist.
41. Vorrichtung nach einem der Ansprüche 23 bis 40, worin die Hilfemaschine so eingerichtet ist, daß sie den Kontext
feststellt, worin der erste Punkt kontextabhängig ist.
42. Vorrichtung nach einem der Ansprüche 23 bis 41, worin der Prozessor so eingerichtet ist, daß er eine grafische
Benutzerschnittstelle implementiert, durch die ein Bediener Sprungkommandos absetzen kann.
43. Vorrichtung nach Anspruch 42, worin der Prozessor so eingerichtet ist, daß er die grafische Benutzerschnittstelle
so implementiert, daß sie Kommandos zur Neupositionierung der Spracherzeugung für den Punkt empfängt.
44. Speichermedium, das Programmanweisungen für die Steuerung eines Prozessors speichert, um das Verfahren
nach einem der Ansprüche 1 bis 22 durchzuführen.
45. Prozessorimplementierbare Programmanweisungen zur Steuerung eines Prozessors zum Ausführen des Verfah-
rens nach einem der Ansprüche 1 bis 22.

40 Revendications

1. Méthode de traitement d'un fichier d'aide hypertextuel de rubriques, au moins certaines des rubriques incluant du
texte et au moins un premier lien hypertextuel vers une rubrique de renvoi, la méthode comprenant les étapes con-
sistant à :
recevoir une demande pour une première rubrique du fichier d'aide hypertextuel de rubriques ;
générer de la parole de manière sérielle pour le texte dans la première rubrique ;
accéder à une rubrique de renvoi spécifiée par ledit lien hypertextuel en réponse à une commande de renvoi
émise par l'utilisateur ; et
générer de la parole de manière sérielle pour le texte dans la rubrique de renvoi ;
caractérisée par l'étape consistant à :
générer de la parole pour le ou chaque lien hypertextuel lorsqu'il est rencontré durant la génération de la parole
sérielle pour le texte et générer un repère sonore de manière à indiquer qu'un lien hypertextuel et non pas du
texte est en train d'être prononcé.
2. Méthode selon la revendication 1, dans laquelle le repère sonore comprend la génération de la parole pour le ou
chaque lien hypertextuel dans une seconde voix qui est différente d'une première voix générée pour le texte dans
la première rubrique ou rubrique de renvoi.

3. Méthode selon la revendication 1 ou 2, incluant l'affichage facultatif du texte et du ou de chaque lien hypertextuel de la rubrique pour laquelle la parole est en train d'être générée.
- 5 4. Méthode selon la revendication 3, dans laquelle les étapes de la revendication 3 sont implémentées par un moteur d'aide, la méthode comprenant de plus les étapes consistant à :

exécuter un programme d'application qui inclut des commandes invocables par l'opérateur pour exécuter le moteur d'aide, ce dernier étant paramétrable pour fournir l'une quelconque des formes d'aide parmi l'aide à parole générée, l'aide affichée, et à la fois l'aide à parole générée et l'aide affichée ; et

10

rendre instantané le moteur d'aide en réponse aux commandes de l'opérateur.
- 15 5. Méthode selon la revendication 4, comprenant de plus l'étape consistant à passer le contexte du programme d'application au moteur d'aide, la parole étant générée pour des rubriques et des rubriques étant facultativement affichées en commençant par le contexte passé à partir du programme d'application.
- 20 6. Méthode selon la revendication 4 ou 5, comprenant de plus l'étape consistant à afficher une interface utilisateur graphique au moyen de laquelle le moteur d'aide est commandé.
7. Méthode selon la revendication 6, dans laquelle le fichier d'aide hypertextuel est configuré de manière hiérarchique, et dans laquelle l'interface utilisateur graphique comporte un jeu de commandes de traitement pour traiter les niveaux hiérarchiques du fichier hypertextuel.
- 25 8. Méthode selon la revendication 7, dans laquelle le jeu de commandes de traitement inclut des commandes de renvoi ou de saut vers des rubriques différentes.
9. Méthode selon la revendication 7, dans laquelle le jeu de commandes de traitement inclut des commandes de renvoi ou de saut vers des niveaux différents.
- 30 10. Méthode selon la revendication 7, dans laquelle le jeu de commandes de traitement inclut des commandes pour repositionner le générateur de parole et pour afficher de manière optionnelle la rubrique.
11. Méthode selon l'une quelconque des revendications précédentes, comprenant de plus l'étape consistant à stocker un premier dudit au moins un lien hypertextuel.
- 35 12. Méthode selon la revendication 11, comprenant de plus l'étape consistant à accéder à la rubrique spécifiée par le lien stocké en réponse à des commandes de renvoi émises par l'utilisateur et à générer de la parole pour la rubrique de renvoi.
- 40 13. Méthode selon la revendication 12, comprenant de plus l'étape consistant à retourner à la rubrique de départ à la conclusion de la rubrique de renvoi.
14. Méthode selon l'une quelconque des revendications précédentes, dans laquelle le fichier hypertextuel est agencé de manière hiérarchique, et comprenant de plus l'étape consistant à détecter et stocker la structure hiérarchique des rubriques.
- 45 15. Méthode selon la revendication 14, comprenant de plus l'étape consistant à effectuer des renvois ou des sauts vers différentes rubriques dans la structure hiérarchique stockée en conformité avec les commandes de l'opérateur.
16. Méthode selon la revendication 14, comprenant de plus l'étape consistant à générer de la parole pour la structure hiérarchique stockée de rubriques.
- 50 17. Méthode selon la revendication 16, comprenant de plus l'étape consistant à effectuer un saut vers l'une sélectionnée des rubriques prononcées.
- 55 18. Méthode selon l'une quelconque des revendications précédentes, comprenant de plus les étapes consistant à, alors qu'on se trouve dans la rubrique de renvoi, accéder à une seconde rubrique de renvoi spécifiée par un autre dudit au moins un lien hypertextuel en réponse à une seconde commande de renvoi émise par l'utilisateur.

19. Méthode selon l'une quelconque des revendications précédentes, comprenant de plus l'étape consistant à retourner à la première rubrique à la conclusion de la génération de la parole pour la rubrique de renvoi.
- 5 20. Méthode selon l'une quelconque des revendications précédentes, comprenant de plus l'étape consistant à détecter le contexte dans lequel la première rubrique est dépendante du contexte.
21. Méthode selon l'une quelconque des revendications précédentes, comprenant de plus l'étape consistant à afficher une interface utilisateur graphique au moyen de laquelle un opérateur peut émettre des commandes de renvoi.
- 10 22. Méthode selon la revendication 21, dans laquelle ladite interface utilisateur graphique reçoit des commandes pour repositionner la génération de la parole pour la rubrique.
23. Appareil pour traiter un fichier d'aide hypertextuel de rubriques, au moins certaines des rubriques incluant du texte et au moins un premier lien hypertextuel vers une rubrique de renvoi ou de saut, l'appareil comprenant :
15 un moyen de réception pour recevoir une demande pour une première rubrique du fichier d'aide hypertextuel de rubriques ;
un moyen de génération de la parole pour générer de manière sérielle de la parole pour le texte dans la première rubrique ; et
20 un moyen d'accès pour accéder à une rubrique de renvoi spécifiée par ledit un lien hypertextuel en réponse à une commande de renvoi reçue de l'utilisateur ;
appareil dans lequel ledit moyen de génération de la parole est adapté pour générer de la parole de manière sérielle pour le texte dans la rubrique de renvoi ; caractérisé en ce que :
25 ledit moyen de génération de la parole est adapté pour générer de la parole pour le ou chaque lien hypertextuel lorsqu'il est rencontré durant la génération de la parole sérielle du texte et pour générer un repère sonore pour indiquer qu'un lien hypertextuel et non pas du texte est en train d'être prononcé.
- 30 24. Appareil selon la revendication 23, dans lequel ledit moyen de génération de la parole est adapté pour générer de la parole pour le ou chaque lien hypertextuel dans une seconde voix qui est différente d'une première voix générée pour le texte dans la première rubrique ou rubrique de renvoi, de manière que le repère sonore indique qu'un lien hypertextuel et non pas du texte est en train d'être prononcé.
- 35 25. Appareil selon la revendication 24, comprenant un processeur pour exécuter des instructions pouvant être mises en oeuvre par processeur, ledit processeur étant adapté pour implémenter des instructions pouvant être mises en oeuvre par processeur afin d'exécuter un programme d'application qui inclut des commandes invocables par l'opérateur pour exécuter un moteur d'aide, ce dernier étant paramétrable pour fournir l'une quelconque des formes d'aide parmi l'aide à parole générée, l'aide affichée, et à la fois l'aide à parole générée et l'aide affichée, ledit moteur d'aide comprenant ledit moyen de réception, ledit moyen de génération de la parole, ledit moyen d'accès,
40 et ledit moyen de commande d'affichage.
26. Appareil selon la revendication 25, dans lequel ledit processeur est adapté pour passer le contexte du programme d'application au moteur d'aide, et le moteur d'aide est adapté pour commencer la génération de la parole pour des rubriques et pour afficher facultativement des rubriques avec le contexte passé à partir du programme d'application.
45
27. Appareil selon la revendication 25 ou 26, dans lequel ledit processeur est adapté pour générer une interface utilisateur graphique pour le contrôle du moteur d'aide.
- 50 28. Appareil selon la revendication 27, dans lequel le fichier d'aide hypertextuel est agencé de manière hiérarchique, et le processeur est adapté pour implémenter l'interface utilisateur graphique afin d'inclure un jeu de commandes de traitement pour traiter les niveaux hiérarchiques des fichiers hypertextuels.
- 55 29. Appareil selon la revendication 28, dans lequel le processeur est adapté pour implémenter l'interface utilisateur graphique afin d'inclure des commandes de renvoi ou de saut vers des rubriques différentes.
30. Appareil selon la revendication 28, dans lequel le processeur est adapté pour implémenter l'interface utilisateur graphique afin d'inclure des commandes de renvoi ou de saut vers des niveaux différents.

31. Appareil selon la revendication 28, dans lequel le processeur est adapté pour implémenter l'interface utilisateur graphique afin d'inclure des commandes pour repositionner la génération de la parole et afficher facultativement la rubrique.
- 5 32. Appareil selon l'une quelconque des revendications 23 à 31, comprenant de plus un moyen pour stocker l'un dudit au moins un lien hypertextuel.
33. Appareil selon la revendication 32, dans lequel ledit moteur d'aide est adapté pour accéder aux rubriques spécifiées par le lien stocké en réponse à des commandes de renvoi émises par l'utilisateur et pour générer de la parole
10 pour la rubrique de renvoi.
34. Appareil selon la revendication 33, dans lequel ledit moteur d'aide est adapté pour retourner vers la rubrique de départ à la conclusion de la rubrique de renvoi.
- 15 35. Appareil selon l'une quelconque des revendications 23 à 34, dans lequel le fichier hypertextuel est agencé de manière hiérarchique, et ledit moteur d'aide est adapté pour détecter et stocker la structure hiérarchique des rubriques.
36. Appareil selon la revendication 35, dans lequel ledit moteur d'aide est adapté pour effectué un saut vers différentes
20 rubriques dans la structure hiérarchique stockée en conformité avec des commandes de l'opérateur.
37. Appareil selon la revendication 36, dans lequel ledit moteur d'aide est adapté pour générer de la parole pour la structure hiérarchique stockée des rubriques.
- 25 38. Appareil selon la revendication 37, dans lequel ledit moteur d'aide est adapté pour effectuer un saut ou un renvoi vers l'une sélectionnée des rubriques prononcées.
39. Appareil selon l'une quelconque des revendications 23 à 38, dans lequel ledit moteur d'aide est adapté pour, alors que l'on se trouve dans la rubrique de renvoi, accéder à une seconde rubrique de renvoi spécifiée par un autre
30 dudit au moins un lien hypertextuel en réponse à une seconde commande de renvoi émise par l'utilisateur.
40. Appareil selon l'une quelconque des revendications 23 à 39, dans lequel ledit moteur d'aide est adapté pour retourner à la première rubrique à la conclusion de la génération de la parole pour la rubrique de renvoi.
- 35 41. Appareil selon l'une quelconque des revendications 23 à 40, dans lequel ledit moteur d'aide est adapté pour détecter le contexte dans lequel la première rubrique est dépendante du contexte.
42. Appareil selon l'une quelconque des revendications 23 à 41, dans lequel ledit processeur est adapté pour afficher une interface utilisateur graphique au moyen de laquelle un opérateur peut émettre des commandes de renvoi ou
40 de saut.
43. Appareil selon la revendication 43, dans lequel ledit processeur est adapté pour implémenter ladite interface utilisateur graphique afin de recevoir des commandes pour repositionner la génération de la parole pour la rubrique.
- 45 44. Support de stockage stockant des instructions de programme pour commander un processeur afin de mettre en oeuvre la méthode selon l'une quelconque des revendications 1 à 22.
45. Instructions de programme implémentables par processeur pour commander un processeur afin de mettre en oeuvre la méthode selon l'une quelconque des revendications 1 à 22.

50

55

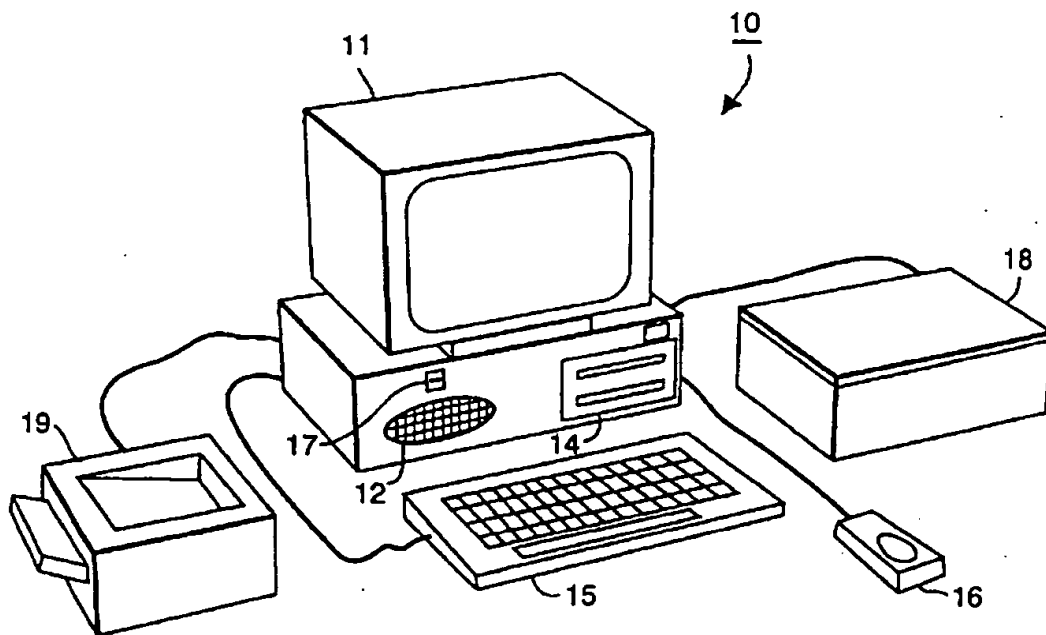


FIG.1

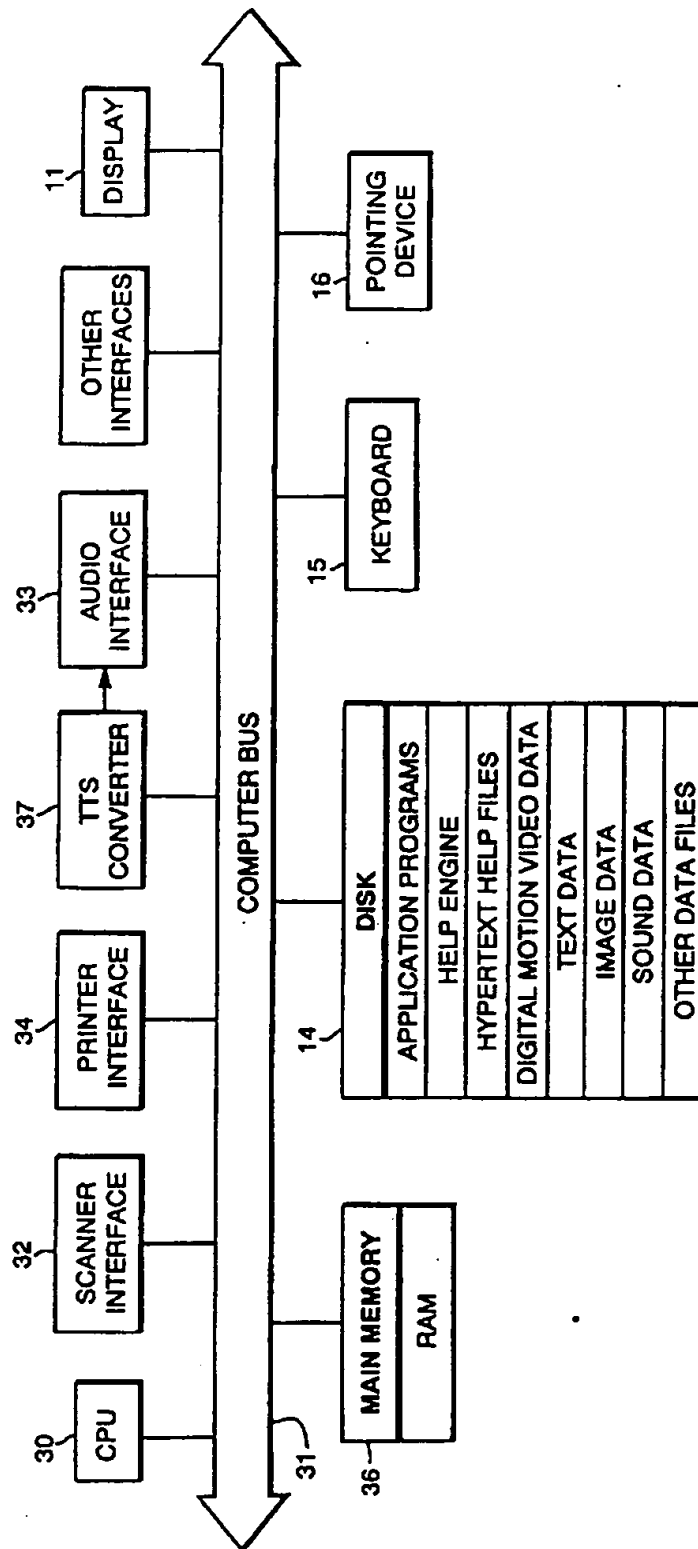


FIG.2

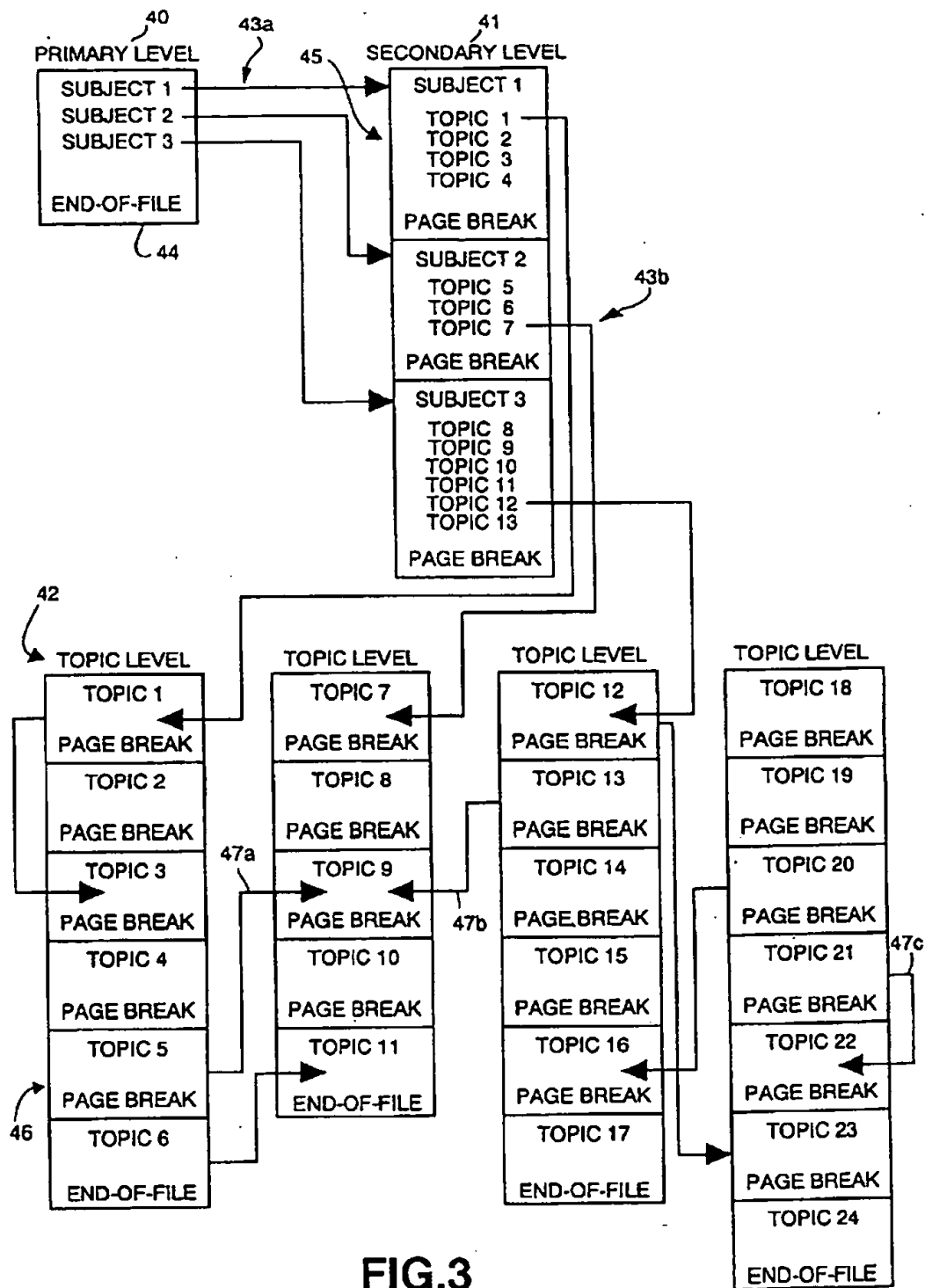


FIG.3

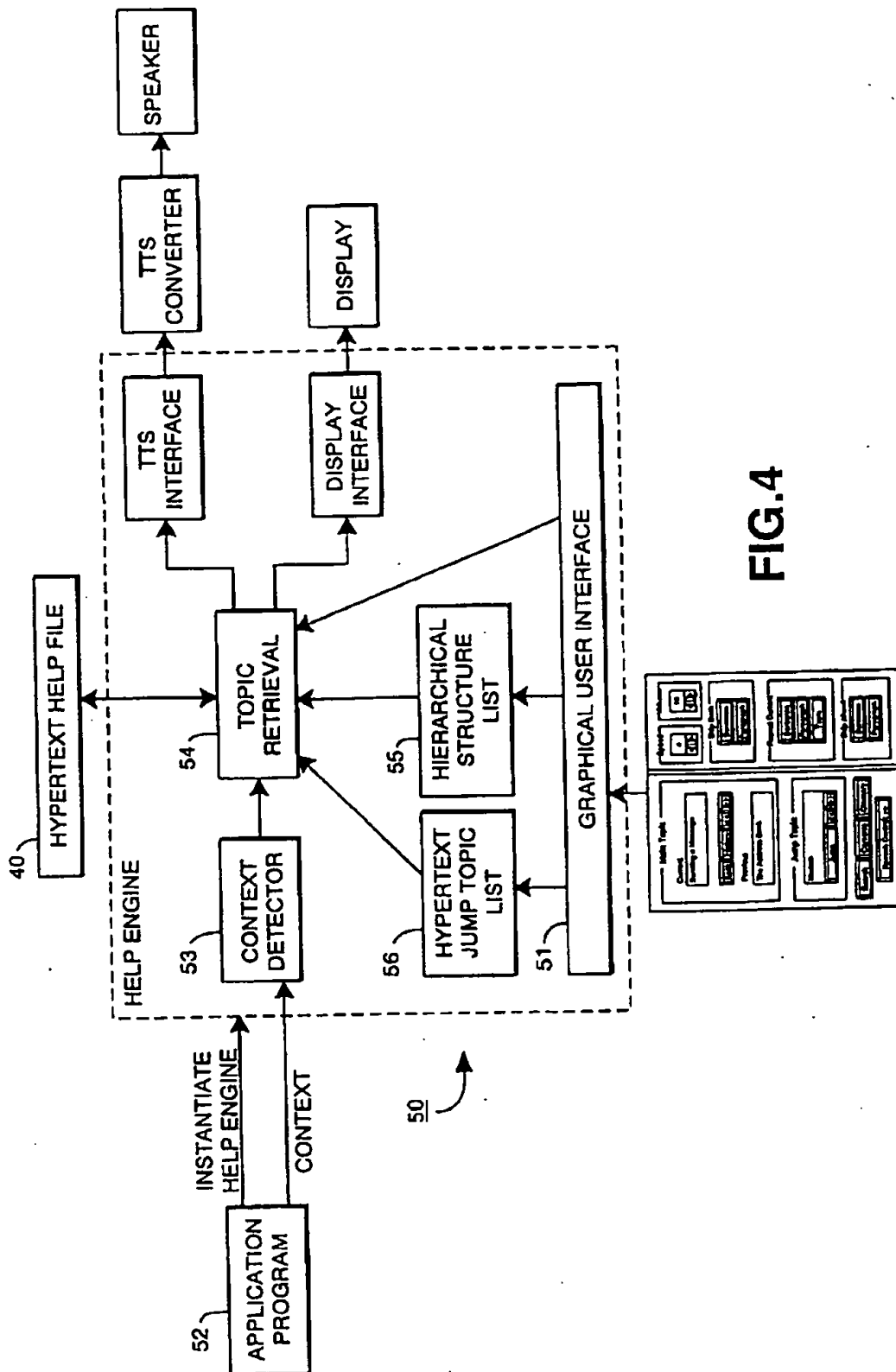


FIG.4

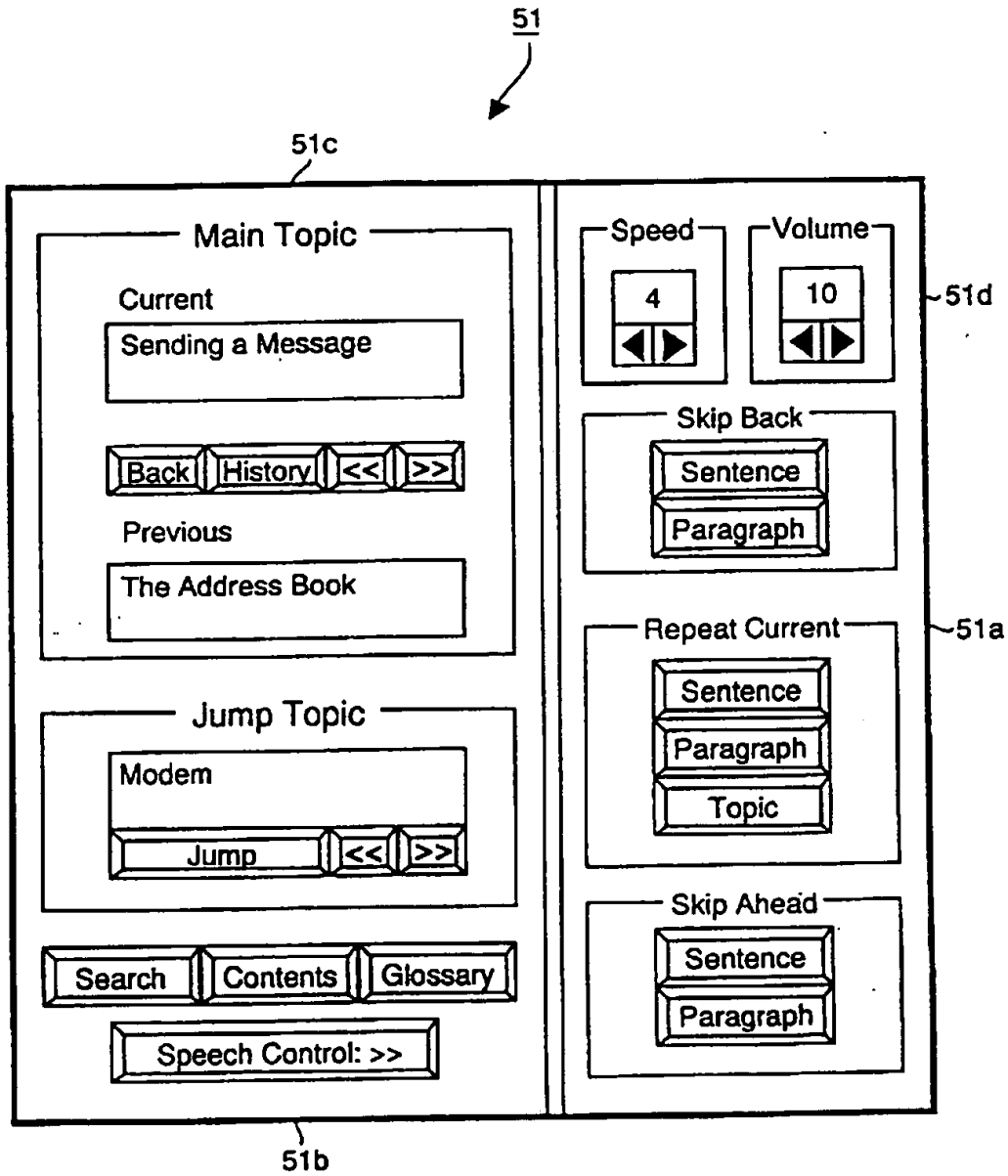


FIG.5

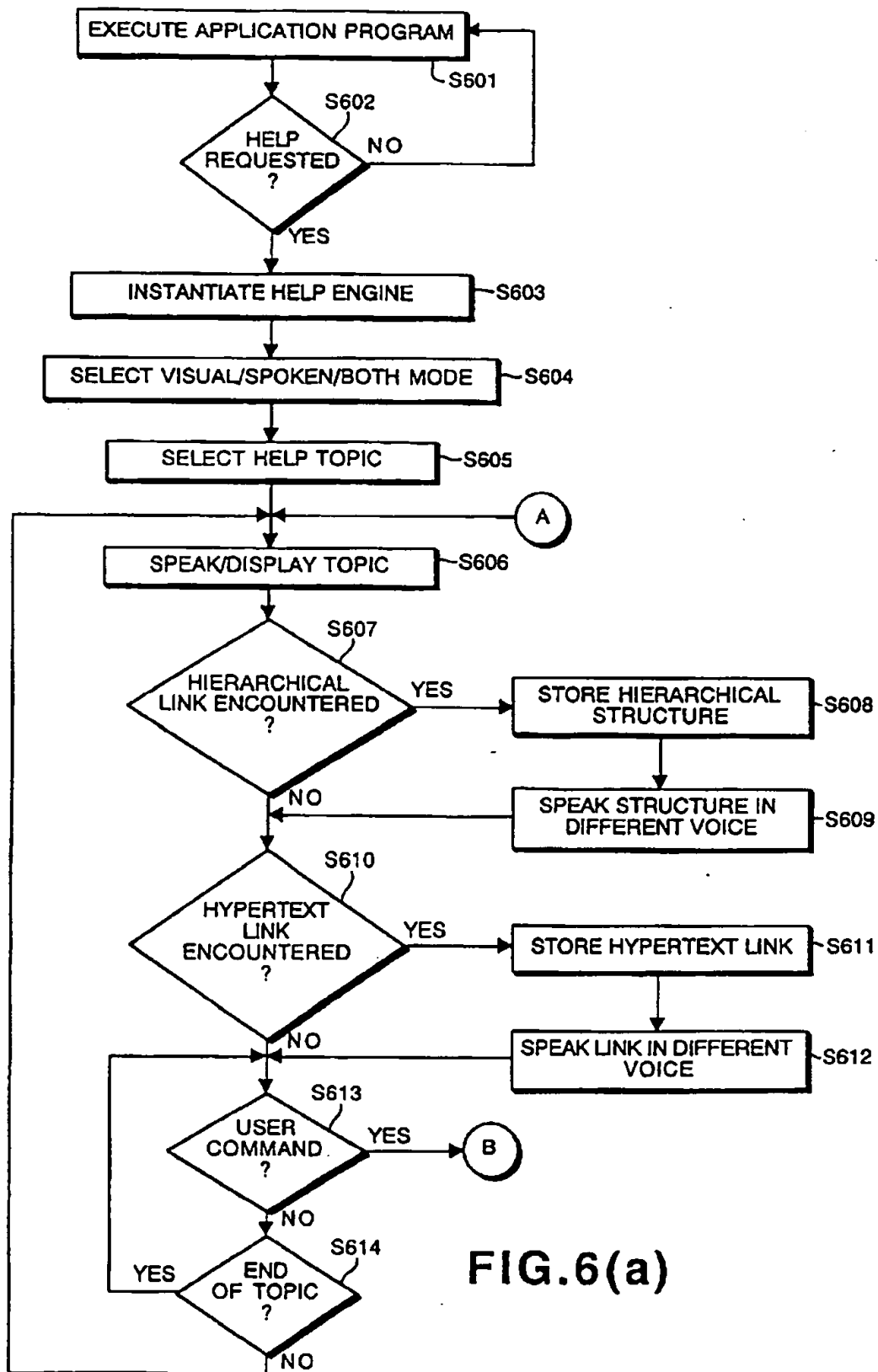


FIG. 6(a)

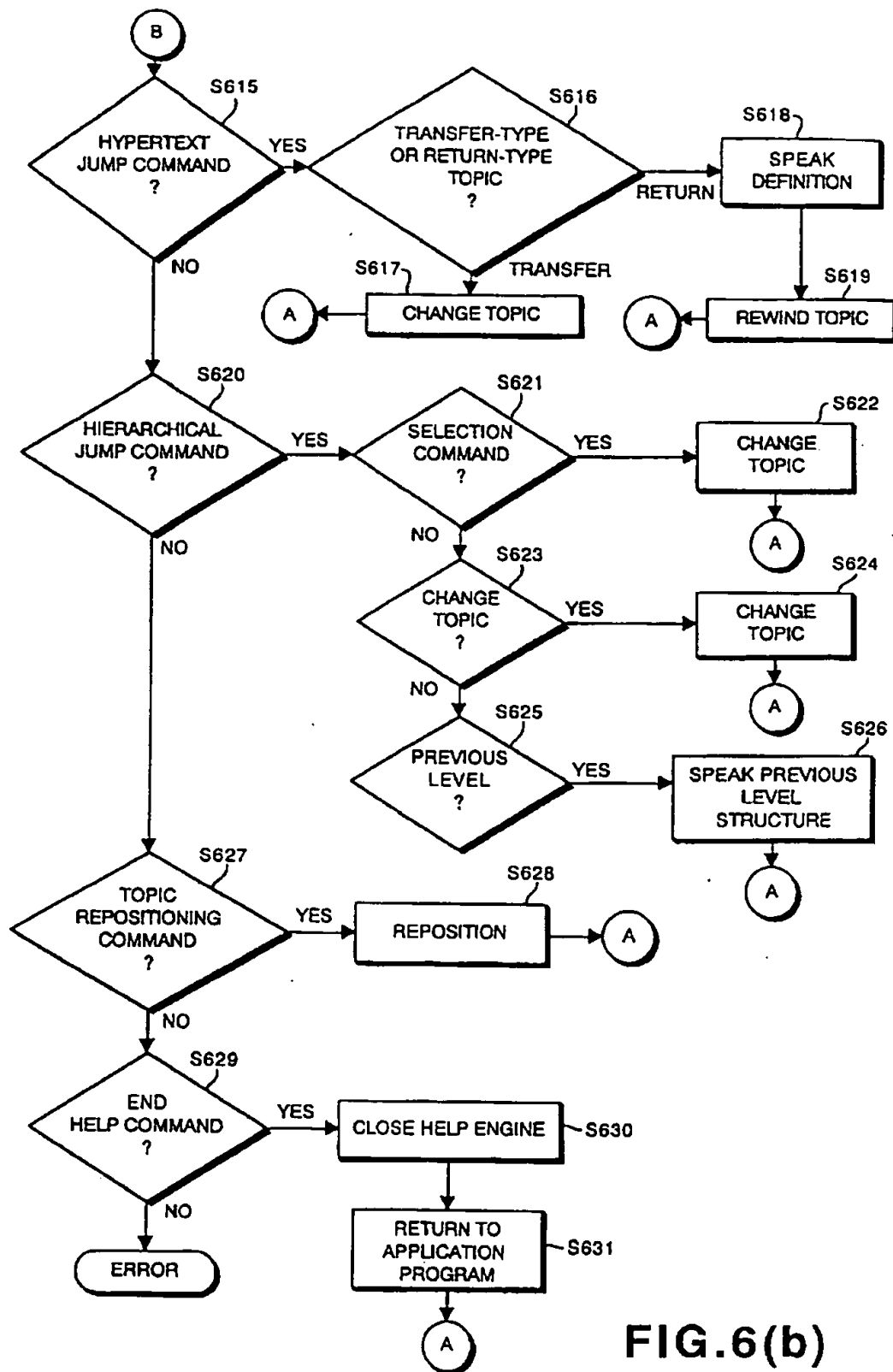


FIG. 6(b)

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.